Listing of claims:

The following listing of claims replaces all previous claim listings in the application:

1. (Previously Presented). A method for managing integrity of a file, the method comprising:

a) at a first time, performing a content checksum of a file in a first repository node to obtain a first checksum and storing the first checksum in the first repository node;

b) at a second time, re-performing the content checksum on the file to obtain a second checksum and comparing the second checksum with the first checksum; and

c) if the second checksum does not equal the first checksum, then at a node with a replica, verifying the replica;

if the replica is verified, then transmitting the verified replica to the first repository node; and

replacing the file with the verified replica;

if the replica is not verified, determining if all other repository nodes with replicas have been checked;

if not, then selecting a node with an alternative replica that has not been verified and verifying the alternative replica;

if the alternative replica is verified, then transmitting the verified alternative replica to the first repository node; and

replacing the file with the verified alternative replica;

if all other repository nodes with replicas have been checked and no verified replicas have been discovered, then determining that file integrity correction failed.

2. (Previously Presented). The method of claim 1 wherein the file is a first file and wherein the method further comprises:

if at least one of a file, a replica, and an alternative replica is verified, then determining if all relevant files have received integrity management;

if all relevant files have received integrity management, then determining that integrity management is complete;

if not, then recursively setting the file equal to the next file and performing (a) –(c) on the next file until all relevant files have received integrity management.

3. (Previously Presented). The method of claim 1 wherein verifying the replica comprises performing a content checksum on the replica to obtain a replica checksum and determining whether the replica checksum equals the first checksum.

4. (Previously Presented). A method for managing integrity of a file, the method comprising:

a) performing a content checksum of a file in a repository node to obtain a checksum and storing the checksum in the repository node;

b) subsequently performing the content checksum on the file to obtain another checksum and comparing the another checksum with the checksum; and

c) if the comparison fails to indicate that the checksums are the same, recovering a copy of the file from another repository node.

5. (Previously Presented). The method of claim 4 wherein recovering a copy of the file from another repository node comprises:

determining repository nodes that have a replica of the file; and

querying the repository nodes.

6. (Previously Presented) The method of claim 5 wherein recovering a copy of the file from another repository node further comprises:

at a node with a replica, verifying the replica;

if the replica is verified, then transmitting the verified replica to the repository node; and

replacing the file with the verified replica.

7. (Previously Presented). The method of claim 6 wherein verifying the replica comprises performing a content checksum on the replica to obtain a replica checksum and determining whether the replica checksum equals the checksum.

8. (Previously Presented). The method of claim 6 wherein recovering a copy of the file from another repository node further comprises:

if the replica is not verified, determining if all other repository nodes with replicas have been checked;

if not, then selecting a node with an alternative replica that has not been verified and verifying the alternative replica;

if the alternative replica is verified, then transmitting the verified alternative replica to the repository node; and

replacing the file with the verified alternative replica.

9. (Original). The method of claim 8 wherein the method further comprises:

if all other repository nodes with replicas have been checked and no verified replicas have been discovered, then determining that file integrity correction failed.

10. (Original). The method of claim 8 wherein the file is a first file and wherein the method further comprises:

if at least one of a file, a replica, and an alternative replica is verified, then determining if all relevant files have received integrity management;

if all relevant files have received integrity management, then determining that integrity management is complete;

if not, then recursively setting the file equal to the next file and performing (a) – (c) on the next file until all relevant files have received integrity management.

11. (Original). The method of claim 4 wherein the checksum is an MD5 checksum.

12. (Previously Presented). A data protection system comprising:

a primary repository node having:

a data mover operative to manage the transfer of data;

a primary repository API in communication with the data mover and operative to communicate with a network;

a primary repository file transfer module in communication with the data mover and operative to receive files;

an integrity service operative to:

perform a content checksum of a file in a repository node to obtain a checksum and to store the checksum in the primary repository node, and

subsequently perform the content checksum on the file to obtain another checksum and to compare the another checksum with the checksum, and

if the comparison fails to indicate that the checksums are the same, output a file recovery request; and

a replicator service in communication with the data mover and the integrity service, the replicator service operative to receive the recovery request from the integrity service and to manage the process of recovering a copy of the file from another repository node.

13. (Previously Submitted). A method for managing integrity of a file, the method comprising

obtaining a checksum of a file at a repository node;

performing a content checksum of a replica of said file to obtain a replica checksum at another repository node;

comparing said replica checksum with said checksum of said file;

if said comparison fails to indicate that said replica checksum and said checksum of said file are the same, performing a content checksum of another replica at a repository node other than said repository node and said another repository node;

if said comparison indicates that said replica checksum and said checksum of said file are the same, transmitting said replica from said another repository node to said repository node.

14. (Previously Submitted). A data protection system, comprising

a repository node coupled to another repository node;

said repository node is configured to

obtain a checksum of said file;

said another repository node is configured to

perform a content checksum of a replica of said file to obtain a replica checksum at another repository node;

compare said replica checksum with said checksum of said file;

if said comparison fails to indicate that said replica checksum and said checksum of said file are the same, perform a content checksum of another replica at a repository node other than said repository node and said another repository node;

if said comparison indicates that said replica checksum and said checksum of said file are the same, transmit said replica from said another repository node to said repository node.